Chapter 1

THE LOGIC OF REUSABLE PROPOSITIONAL OUTPUT WITH THE FULFILMENT CONSTRAINT

Leendert van der Torre
Department of Artificial Intelligence
Vrije Universiteit
De Boelelaan 1081a, 1081 HV, Amsterdam
The Netherlands
torre@cs.vu.nl

Abstract This paper shows the equivalence of three ways of expressing a certain strong consistency constraint – called the fulfilment constraint – on proofs of the logic of reusable propositional output: as a global requirement on proofs, as a local requirement on labels of formulas, and by phasing of proof rules. More specifically, we first show that the fulfilment constraint may be expressed either as a requirement on the historical structure of the proof tree or as a requirement on the contents of labels attached to its nodes. Second, we show that labelled proofs may be rewritten into a tightly phased form in which rules are applied in a fixed order. Third, we show that when a proof is in such a phased form, the consistency check on labels becomes redundant.

Keywords: input-output logics, qualitative decision theory, deontic logic

1. INTRODUCTION

In this paper we consider the logic of reusable propositional output [5] with a strong consistency constraint called the fulfilment constraint. It has the following two characteristic properties.

The identity 'if input a then output a' is *not* a theorem. To avoid confusion with conditional or counterfactual logic we therefore write (a, x) for 'if input a then output x' instead of $a \rightarrow x$ or $a > x$. The logic has the usual rules of strengthening of the input, weakening of the output, cumulative transitivity or cut, the conjunction rule for the output and the disjunction rule for the input.

Proofs are constrained in the sense that for each case each derived conditional can be achieved or fulfilled together with the conditionals it is derived from. The restriction to cases – as we explain in detail later – enables complex forms of reasoning by cases.

Consistency constraints usually lead to non-monotonic behavior. However, no close relationship to well-known non-monotonic logics is to be expected, not only because the underlying logic (of reusable output) is different from classical logic, but also because the form of our fulfilment constraint is different from consistency constraints used in non-monotonic reasoning.

The fulfilment constraint was originally proposed to formalize preferences, desires and goals of a planning agent in a decision-theoretic context [8, 9, 10], though it can also formalize for example obligations and beliefs in normative and epistemic contexts. Desirable properties for this decision-theoretic context are that the logic validates strong proof rules like strengthening of the input and transitivity, and that it – as we again explain in detail later – supports context-sensitive and conflict-tolerant reasoning. As a reminder of this original motivation we refer to the premise set of conditionals as the goal base or G.

This paper shows the equivalence of three ways of expressing the fulfilment constraint on derivations in the logic of reusable propositional output: as a global requirement on derivations, as a local requirement on labels, and by phasing. The second and third way are implemented in a labelled deductive system [2]. Labelled conditionals $(a, x)_L$ can *roughly* be read as 'if a then x, against the proof background of L.' The label L contains information about the context in which the goal is derived.

- In the second way, the label L keeps track of the *conditionals* used to derive $(a, x)_L$. The global consistency constraint on fulfilments of conditionals in proofs is shown to be equivalent to a local consistency constraint on fulfilments of conditionals in this label.
- In the third way, the label L keeps track of the *proof rules* used to derive $(a, x)_L$. Each proof rule is associated with a so-called phase, an integer, and a proof rule may no longer be used once a proof rule from a higher phase has been applied.

The equivalence between the global and local constraint is not surprising, in contrast to their equivalence to phasing. The latter can also be expressed as follows. In the logic of reusable propositional output with the fulfilment constraint, the conditional (a, x) can be derived from the conditional base G if and only if there is a phase-4 conditional (a, x) of G, which is stepwise defined as follows. Ignoring replacements by logical equivalents for input and output, the conditional (a, x) is

a phase-1 conditional of G if $a \wedge x$ is consistent and there is a conditional $(b, x) \in G$ such that a classically implies b, i.e. $a \vdash b$;

a phase-2 conditional of G if there is a set of phase-1 conditionals of G $\{(a, x_1), (a \wedge x_1, x_2), \ldots, (a \wedge x_1 \wedge \ldots \wedge x_{n-1}, x_n)\}$ such that x is classically equivalent with $x_1 \wedge \ldots \wedge x_n$, i.e. $x \dashv\vdash x_1 \wedge \ldots \wedge x_n$;

a phase-4 conditional of G if there is a set of phase-2 conditionals of G $\{(a_1, x_1), \ldots, (a_n, x_n)\}$ such that $a \dashv\vdash a_1 \vee \ldots \vee a_n$ and $x_i \vdash x$ for all $i = 1 \ldots n$.

This paper is organized as follows. We first introduce the logic of reusable propositional output (Section 2.) and the fulfilment constraint as a global requirement (Section 3.), and we illustrate them by several examples. Then we introduce the labelled deductive system and we show that the fulfilment constraint may be expressed indifferently as a requirement on the historical structure of the proof tree or as a requirement on the contents of labels attached to its nodes (Section 4.). Thereafter we show that labelled proofs may be brought into a tightly phased form in which rules are applied in a fixed order, and when a proof is in such a phased form, the consistency check on the labels becomes redundant (Section 5.). Finally, we illustrate phasing by reconsidering the examples (Section 6.).

2. REUSABLE PROPOSITIONAL OUTPUT

David Makinson (private communication) introduced the concept of 'reusable output' to formalize a black box, in which we may feed propositions, and that produces propositions as output, with every element of output available for re-use as input. For example, classical consequence $\vdash$ may be seen in this way. Every element of the output may be recycled as input, for it satisfies the rule known as cumulative transitivity or cut, that if $a \vdash x$ and $a \wedge x \vdash y$ then $a \vdash y$. This tells us that output x may be used as input alongside a and anything thus obtained is considered to be an output of a itself. Classical consequence, as well as other conditional logics satisfying the cut rule, has a further feature: inputs are themselves outputs, since $a \vdash a$ for any consequence relation. However, in certain contexts this is undesired. For example, in the logic of obligations – called deontic logic – cumulative transitivity – called (cautious) deontic detachment – is desired, but deriving is from ought or vice versa is undesired. In [5] the following logic of reusable propositional output has been introduced.[1]

Definition 1 (The logic of reusable propositional output) *The language consists of conditionals (a, x), with (input) a and (output) x sentences of a propositional base logic $\mathcal{L}$. We say that a conditional (a, x) is consistent if and only if $a \wedge x$ is consistent in $\mathcal{L}$, and a set of conditionals or conditional base is consistent if and only if each of its elements is. The consistent conditional base G implies (a, x) in the logic of reusable propositional output, written as $G \vdash_{\mathrm{RO}} (a, x)$, if and only if (a, x) can be derived from G with replacements by logical equivalents (for input and output) and the following five rules.*

$$\mathrm{SI}: \frac{(a,x)}{(a \wedge b, x)} \qquad \mathrm{CT}: \frac{(a,x), (a \wedge x, y)}{(a,y)} \qquad \mathrm{AND}: \frac{(a,x), (a,y)}{(a, x \wedge y)}$$

$$\mathrm{WO}: \frac{(a,x)}{(a, x \vee y)} \qquad \mathrm{OR}: \frac{(a,x), (b,x)}{(a \vee b, x)}$$

We give two examples of typical derivations of the logic. The first derivation illustrates how cumulative transitivity together with strengthening of the input formalizes that conditional rules can be applied one after the other. Derivations go 'as far as possible.'

$$\dfrac{\dfrac{(a,b) \quad \dfrac{(b,c)}{(a \wedge b, c)}\,\mathrm{SI}}{(a,c)}\,\mathrm{CT} \qquad \dfrac{(c,d)}{(a \wedge c, d)}\,\mathrm{SI}}{(a,d)}\,\mathrm{CT}$$

The second derivation (taken from [4]) illustrates how the disjunction rule supports reasoning by cases, where we write $\top$ for any tautology like $p \vee \neg p$.

$$\dfrac{\dfrac{(a, x \wedge y)}{(a,x)}\,\mathrm{WO} \qquad \dfrac{(\neg a, x \wedge \neg y)}{(\neg a, x)}\,\mathrm{WO}}{(\top, x)}\,\mathrm{OR}$$

The basic idea of the additional consistency constraints on proofs of the logic of reusable output is to block derivations from conflicting conditionals, without blocking the latter derivation for reasoning by cases.

[1] In [5] the logic of reusable propositional output also satisfies the axiom $(a, \top)$ to facilitate the semantic presentation. Since we do not consider semantics in this paper, we do not consider this axiom either.

3. A GLOBAL CONSTRAINT ON PROOFS

To deal with reasoning by cases, the OR rule is thought of as creating different cases or parallel dependency tracks. Given a node in a derivation tree, we travel upwards the derivation tree generated by the node, splitting in two cases at every application of the OR rule, such that one case contains one of the two premise nodes and the other case contains the other premise node.

Definition 2 (Cases) *Consider a tree-like proof of a conditional (the root) from a set of premises (the leaves). A case or dependency tree of a proof is represented by a set of nodes from the tree. The set of cases of the proof can be computed with the following procedure.*

1. *Initially, the set C contains one element, the set with the root.*
2. *As long as there is a non-leave node in some set of C without one of its parents, then either:*
   - *if the node is not derived by* OR*, then add its parents to the set;*
   - *otherwise create two copies of the set, one for each of its parents; add this parent.*
3. *Each set of nodes in C represents a case of the proof.*

In [5] three different options for additional consistency constraints on derivations have been discussed. They are either based on materializations $(a \rightarrow x)$, outputs (x) or fulfilments $(a \wedge x)$ of the nodes (a, x) of a case. It is suggested that the materialization constraint is best suited to formalize normative reasoning, due to its relation with violations: (a, x) is violated if its materialization $a \rightarrow x$ is false and thus $a \wedge \neg x$ is true. In this paper we only consider the latter most cautious fulfilment constraint. As argued extensively in [12, 14, 15] we think that this constraint is best suited for decision-theoretic reasoning, because it not only refers to qualitative abstractions of penalties of violations but also of rewards of fulfilments.

Definition 3 (The fulfilment constraint) *A consistent conditional base G implies (a, x) in the logic of reusable propositional output with the fulfilment constraint, written as $G \vdash_{\mathrm{ROF}} (a, x)$, if and only if $G \vdash_{\mathrm{RO}} (a, x)$ and for each case of this proof $\{(b_1, y_1), \ldots, (b_m, y_m)\}$ we have that $b_1 \wedge y_1 \wedge \ldots \wedge b_m \wedge y_m$ is consistent in $\mathcal{L}$.*

Due to the consistency constraint, the inference relation is no longer idempotent. That is, if we write $out(G)$ for $\{(a, x) \mid G \vdash_{\mathrm{RO}} (a, x)\}$ and $outF(G)$ for $\{(a, x) \mid G \vdash_{\mathrm{ROF}} (a, x)\}$, then we have $out(G) = out(out(G))$ because out is a closure operator, but we do not have $outF(G) = outF(outF(G))$. The inference relation $\vdash_{\text{ROF}}$, just like $\vdash_{\text{RO}}$, is monotonic ($outF(G) \subseteq outF(G \cup G')$) and compact ($(a,x) \in outF(G)$ implies $(a,x) \in outF(G')$ for some finite $G' \subseteq G$).

The fulfilment consistency constraint has originally been proposed to formalize preferences, desires and goals of a planning agent [8, 9, 10]. Two important properties for this decision-theoretic context are that the logic with this constraint supports context-sensitive and conflict-tolerant reasoning. First, context-sensitive reasoning – called contrary-to-duty reasoning in a normative context [13] – refers to the situation in which a conditional has been 'violated', i.e. in which its materialization $a \rightarrow x$ is false. We say that (b,y) is a secondary conditional of the primary conditional (a,x) if and only if $b \wedge (a \rightarrow x)$ is classically inconsistent. Due to the consistency check, primary and secondary conditionals cannot be combined in a proof, nor can one be derived from the other. For example, consider a decision-theoretic variant of the so-called Chisholm paradox [1]. A robot may have the goal 'to get me some coffee' $(\top, c)$, 'to inform me that coffee will arrive if it gets me some coffee' (c,i), and 'not to inform me that coffee will arrive if it does not get me coffee' $(\neg c, \neg i)$. The secondary conditional $(\neg c, \neg i)$ refers to the 'no coffee' context in which the primary conditional is violated. The coffee machine may be broken, or – if the robot is autonomous – it may decide that other actions have a higher priority. The consistency check blocks the second step of the following counterintuitive derivation of the goal 'to inform me that coffee will arrive if it does not get me coffee' in the unrestricted logic of reusable propositional output.

$$\cfrac{\cfrac{\cfrac{(\top,c) \quad (c,i)}{(\top,i)}\,\text{CT}}{(\neg c,i)}\,\text{SI} \qquad (\neg c,\neg i)}{(\neg c, i \wedge \neg i)}\,\text{AND}$$

Second, the logic supports reasoning with conflicting preferences, for example due to conflicting objectives. The consistency check ensures that we have $(\top,x),(\top,\neg x) \not\vdash_{\text{ROF}} (\top, x \wedge \neg x)$ and $(\top,x),(\top,\neg x) \not\vdash_{\text{ROF}} (\top,y)$ if y is not logically implied by x or $\neg x$. In particular, the second derivation step in the following counterintuitive derivation in the unrestricted logic of reusable propositional output is blocked.

$$\cfrac{\cfrac{\cfrac{(\top,x)}{(\top,x \vee y)}\,\text{WO} \qquad (\top,\neg x)}{(\top, y \wedge \neg x)}\,\text{AND}}{(\top,y)}\,\text{WO}$$

The latter derivation also illustrates why the logic has a global consistency constraint instead of a local one at each derivation step. Each conditional in the proof is consistent, but still counterintuitive conclusions can be derived. Finally, notice that the following derivations – so-called forbidden conflict, proof by contradiction and contraposition – would be blocked by the consistency constraint if they were valid in the underlying logic. The first two combine conditionals with contradictory fulfilments and the latter derives a conditional whose input is inconsistent with the fulfilment of the premise.

$$\text{FC} : \frac{(a,x),(a\wedge b,\neg x)}{(a,\neg b)} \qquad \text{PBC} : \frac{(a,\bot)}{(\top,\neg a)} \qquad \text{CP} : \frac{(a,x)}{(\neg x,\neg a)}$$

In the following section we formalize the fulfilment consistency constraint as a local requirement on labels in a labelled deductive system, and we show that the global requirement on proofs and the local requirement on labels are equivalent.

4. A LOCAL CONSTRAINT ON LABELS

The examples in the previous section raise the question whether the concept of reusable output can be combined with context sensitivity in a logic that is – besides monotonic and compact – also idempotent. This combination (without conflict tolerance) has been discussed during three decades of research in deontic logic, following the publication of Chisholm's paradox [1] and several other notorious deontic paradoxes, and more recently it has been discussed (together with conflict tolerance) in qualitative decision theory. Several approaches have been proposed, see e.g. [13] for a survey, but no consensus has been reached. The derivation of $(\neg c, i)$ from $(\top, c)$ and (c, i) from Chisholm's set in the previous section indicates that we have to extend the language or introduce other machinery to combine cumulative transitivity with strengthening of the input. In [13] we therefore propose to formalize reusable output by the following adapted cumulative transitivity rule: if we have (a, x) and $(a \wedge x, y)$ then we can derive $(a, x \wedge y)$ instead of (a, y). This rule can be combined with strengthening of the input (conflict tolerant) and extended with the conjunction rule (conflict intolerant) by adding a local consistency constraint in the following way.[2]

Definition 4 (Logic of imperatives) *The consistent conditional base G implies (a, x) in the logic of imperatives, written as $G \vdash_{\mathrm{I}} (a, x)$, if and only*

[2]The deontic logic proposed in [13] differs from the logic of imperatives in this paper in several aspects. For example, it satisfies the axiom (a, a) to facilitate the semantic (preference-based) presentation, and it satisfies the so-called 'deontic' axiom $\neg(a, \bot)$, where $\bot$ stands for any contradiction like $p \wedge \neg p$, to express that 'ought implies can.' We do not consider the former because in this paper we are not interested in semantics, and we do not consider the latter because we cannot express this axiom in our simple formal language. Moreover, this deontic logic has the conjunction rule for the output, but it only has a restricted strengthening of the input rule to deal with a particular problem of reasoning about normative conflicts: the consistency of 'the window should be open if the sun shines' and 'it should be closed if it rains' [16].

if (a, x) can be derived from G with replacements by logical equivalents (for input and output) and the following two rules, extended with the following condition R_{IO} with a consistency constraint on the conjunction of the input and output.

R_{IO}**:** *(a, x) may only be derived if $a \wedge x$ is consistent: it must always be possible to fulfill a derived conditional.*

$$\text{SI}_{R_{IO}} : \frac{(a,x), R_{IO}}{(a \wedge b, x)} \qquad \text{CTA}_{R_{IO}} : \frac{(a,x), (a \wedge x, y), R_{IO}}{(a, x \wedge y)}$$

Moreover, this minimal logic of imperatives can be extended with the unrestricted conjunction rule.

$$\text{AND} : \frac{(a,x), (a,y)}{(a, x \wedge y)}$$

Obviously the logic of imperatives is a closure operator and thus monotone, compact and idempotent. Moreover, reconsider our variant of Chisholm's paradox with the robot's goals 'to get me some coffee' $(\top, c)$, 'to inform me that coffee will arrive if it gets me some coffee' (c, i), and 'not to inform me that coffee will arrive if it does not get me coffee' $(\neg c, \neg i)$. In the logic of imperatives the goal 'to get me some coffee and inform me about it' $(\top, c \wedge i)$ can be derived, but $(\top, i)$ and $(\neg c, i)$ cannot be derived.

$$\frac{(\top, c) \quad (c, i)}{(\top, c \wedge i)} \text{ CTA}$$

The logic of imperatives cannot be extended with WO, because the counterintuitive derivation of Chisholm's paradox would directly be reinstated, and the extension with OR also leads to counterintuitive consequences. However, these two derivations are in most cases intuitive. For example, in a decision-theoretic context WO can be used to infer from a goal for $x \wedge y$ that there are two subgoals, one for x and one for y. Moreover, the OR rule can be used to reason by cases. The formal language has therefore been extended with labels [11, 4, 8, 9, 10] in versions of a labelled deductive system as it was introduced by Gabbay in [2]. This extension can be based on materializations, outputs or fulfilments, just like the consistency constraints in the logic of reusable output. Again we only consider the latter most cautious option.

In the following definition we introduce the first labelled deductive system of this paper. Roughly speaking, the label L of a conditional $(a, x)_L$ consists of a record of the fulfilments (F) of the premises that are used in the derivation of $(a, x)_L$. Where there is no application of reasoning by cases, F can be taken to be a boolean formula, that grows by conjunction as premises are combined. But in general, to cover the parallel tracks created through reasoning by cases, we need to consider sets of boolean formulas [4]. Each formula $(a,x)_L$ occurring as a premise has a label that consists of its own (propositionally consistent) fulfilment. The label of a derived formula is the union (OR) or the product (SI, CTA) of the fulfilments of the premises used in this inference rule, where the product is defined by

$$\{a_1,\ldots,a_n\} \times \{b_1,\ldots,b_m\} = \{a_1 \wedge b_1,\ldots,a_1 \wedge b_m,\ldots,a_n \wedge b_m\}$$

The labels are used to check that fulfilments are consistent.

Definition 5 (Labelled deductive system with fulfilment constraint) *Let $\mathcal{L}$ be a propositional base logic. The language of the labelled system consists of the labelled conditionals $(a,x)_L$, with a and x sentences of $\mathcal{L}$, and L a set of sentences of $\mathcal{L}$ (fulfilments). A premise of the labelled system is a formula $(a,x)_{\{a\wedge x\}}$, where $a \wedge x$ is consistent in $\mathcal{L}$. The labelled deductive system with the fulfilment constraint consists of the inference rules below, extended with the following condition R_F.*

R_F**:** *$(a,x)_F$ may only be derived if each $f \in F$ is consistent: it must always be possible to fulfill a derived conditional and each of the conditionals it is derived from, though – for reasoning by cases – not necessarily all of them at the same time.*

The inference rules of the labelled system are replacements by logical equivalents (for input and output) and the following four rules.

$$\mathrm{SI}_{R_F} : \frac{(a,x)_F, R_F}{(a \wedge b, x)_{F\times\{b\}}} \qquad \mathrm{CTA}_{R_F} : \frac{(a,x)_{F_1}, (a\wedge x, y)_{F_2}, R_F}{(a, x \wedge y)_{F_1 \times F_2}}$$

$$\mathrm{WO}_{R_F} : \frac{(a,x)_F, R_F}{(a, x \vee y)_F} \qquad \mathrm{OR}_{R_F} : \frac{(a,x)_{F_1}, (b,x)_{F_2}, R_F}{(a \vee b, x)_{F_1 \cup F_2}}$$

We say $(a_1,x_1),\ldots,(a_n,x_n) \vdash_{\mathrm{LDSF}} (a,x)$ if there is a conditional $(a,x)_L$ that can be derived from $\{(a_1,x_1)_{\{a_1\wedge x_1\}},\ldots,(a_n,x_n)_{\{a_n \wedge x_n\}}\}$.

To show the equivalence between the logic of reusable propositional output with the fulfilment constraint and the labelled deductive system, we have to show that they validate the same proof rules and that it is equivalent to check at each node of the proof tree (as in the labelled deductive system) or only at the root (as in the logic). We therefore prove several lemmas. The first one shows that the logic implies the counterparts of the proof rules of the labelled system.

Lemma 6 *The following rule is implied by the logic of reusable output.*

$$\mathrm{CTA} : \frac{(a,x), (a \wedge x, y)}{(a, x \wedge y)}$$

Proof We have the following derivation.

$$\dfrac{(a,x) \qquad \dfrac{(a,x) \quad (a \wedge x, y)}{(a,y)}\,\text{CT}}{(a, x \wedge y)}\,\text{AND}$$

The fulfilment constraint holds this derivation if it holds for CTA. ■

Before we show that the labelled deductive system implies the proof rules of the logic, we prove the following useful lemma.

Lemma 7 *For each conditional $(a,x)_F$ derived in the labelled deductive system we have that each element of F classically implies $a \wedge x$.*

Proof By induction on the structure of the proof tree. The property trivially holds for the premises, and it is easily seen that the proof rules retain the property. ■

The third lemma shows that the labelled deductive system, and in particular the unusual rule CTA, implies counterparts of the proof rules of the logic, such as the cumulative transitivity rule and the conjunction rule.

Lemma 8 *The following cumulative transitivity rule and conjunction rule are implied by the labelled deductive system with the fulfilment constraint.*

$$\text{CT}_{R_F} : \frac{(a,x)_{F_1}, (a \wedge x, y)_{F_2}, R_F}{(a,y)_{F_1 \times F_2}} \qquad \text{AND}_{R_F} : \frac{(a,x)_{F_1}, (a,y)_{F_2}, R_F}{(a, x \wedge y)_{F_1 \times F_2}}$$

Proof We have the following derivations.

$$\dfrac{\dfrac{(a,x)_{F_1} \quad (a \wedge x, y)_{F_2}}{(a, x \wedge y)_{F_1 \times F_2}}\,\text{CTA}}{(a,y)_{F_1 \times F_2}}\,\text{WO} \qquad \dfrac{(a,x)_{F_1} \quad \dfrac{(a,y)_{F_2}}{(a \wedge x, y)_{F_2 \times \{x\}}}\,\text{SI}}{(a, x \wedge y)_{F_1 \times F_2 \times \{x\}}}\,\text{CTA}$$

Moreover, from Lemma 7 follows that in the second derivation each element of F_1 implies x, and therefore $F_1 \times F_2 \times \{x\}$ is equivalent with $F_1 \times F_2$. From the consistency of each element of $F_1 \times F_2$ (root of the left side derivation) follows the consistency of each element of $F_1 \times F_2 \times \{x\}$ (root of the right side derivation) and the consistency of each element of $F_2 \times \{x\}$ (other node of the right side derivation). ■

The fourth lemma below shows – as the last step of the previous proof already suggested – that it is equivalent to check the consistency of the labels at each node or only at the root.

Lemma 9 *Consider any potential derivation of the labelled system, not necessarily satisfying R_F. Then the following two conditions are equivalent:*

1. *The final derivation satisfies condition R_F,*
2. *The derivation satisfies R_F everywhere.*

Proof The labels are, in a suitable sense, cumulative. Every element of every label in the derivation is classically implied by some element of the label of the final conclusion. ■

The first theorem shows the equivalence between the global constraint on proofs and the local constraint on labels of formulas.

Theorem 10 (Equivalence ROF and LDSF) *Let G be a consistent conditional base. We have $G \vdash_{\text{ROF}} (a, x)$ if and only if $G \vdash_{\text{LDSF}} (a, x)$.*

Proof For each proof of (a, x) from G in one system, we can give a proof in the other system. This proof contains two parts. First we show how a proof tree of the second system can be constructed from the proof tree of the first system, and then we show that if the fulfilment constraint holds for the first proof, then it must also hold for the second proof.

$\Rightarrow$ Given a proof of (a, x) in the logic of reusable output with the fulfilment constraint. To construct a proof in the labelled deductive system, first replace occurrences of CT and AND by equivalent proofs using SI, CTA and WO, as indicated in Lemma 8. Then label the premises with the conjunction of their input and output, and label the other nodes as indicated by the rules of the labelled system. We now show that if the consistency constraint holds for the first proof, then it also holds for the second proof. First, the consistency of all cases implies the consistency for all elements of the label of the root. Second, the consistency of all elements of the label of the root implies the consistency of all elements of the label of each node, see Lemma 9.

$\Leftarrow$ Given a proof of (a, x) in the labelled deductive system. To construct a proof in the logic of reusable output, first replace occurrences of CTA by equivalent proofs using CT and AND, see Lemma 6, and then remove all labels. The consistency of each element of the label of the root in the first proof implies the consistency of a case, because each element of F in the label implies the fulfilments of all nodes of a case, and we are done. ■

The first theorem and its proof has been instructive but relatively straightforward. We now proceed with the third way of expressing the fulfilment constraint.

5. PHASING OF PROOF RULES

There is another way in which the logic of imperatives in the previous section has been extended to incorporate the proof rules WO and OR: by so-called phasing of proof rules [6, 7, 9].[3] The idea of phasing is that the proof rules can only be applied in a restricted order. The counterintuitive derivations cannot be derived when SI and CTA cannot be applied once either WO or OR has been applied. In this section the strict order SI, CTA, WO and OR is implemented in the second version of a labelled deductive system.

The allowed order of application is represented by a phasing function ρ, defined by $\rho(\mathrm{SI}) = 1$, $\rho(\mathrm{CTA}) = 2$, $\rho(\mathrm{WO}) = 3$ and $\rho(\mathrm{OR}) = 4$. The label L of a premise $(a, x)_L$ is 0, and the label of a derived formula is the phase of the last proof rule used to derive (a, x). Thus, the label represents a record of the proof rules that are used in the derivation of $(a, x)_L$. The labels are used to check that the phase of reasoning is non-decreasing.

Definition 11 (Labelled deductive system with phasing) *Let $\mathcal{L}$ be a propositional base logic and let ρ be the phasing function from proof rules to integers defined by $\rho(\mathrm{SI}) = 1$, $\rho(\mathrm{CTA}) = 2$, $\rho(\mathrm{WO}) = 3$ and $\rho(\mathrm{OR}) = 4$. The language of the labelled system consists of the labelled conditionals $(a, x)_L$, with a and x sentences of $\mathcal{L}$, and L an integer (the phase). A premise is a formula $(a, x)_0$, where $a \wedge x$ is consistent in $\mathcal{L}$. The labelled deductive system with phasing consists of the inference rules below, extended with the following condition $R_{IOp} = R_{IO} + R_p$.*

R_{IO}: *$(a, x)_p$ may only be derived if $a \wedge x$ is consistent: it must always be possible to fulfill a derived conditional.*

R_p: *$(a, x)_p$ may only be derived if $p \geq p_i$ for all $(a_i, x_i)_{p_i}$ it is derived from.*

The inference rules of the labelled system are replacements by logical equivalents (for input and output) and the following four rules.

$$\mathrm{SI}_{R_{IOp}} : \frac{(a,x)_p, R_{IOp}}{(a \wedge b, x)_{\rho(\mathrm{SI})}} \qquad \mathrm{CTA}_{R_{IOp}} : \frac{(a,x)_{p1}, (a \wedge x, y)_{p2}, R_{IOp}}{(a, x \wedge y)_{\rho(\mathrm{CTA})}}$$

$$\mathrm{WO}_{R_{IOp}} : \frac{(a,x)_p, R_{IOp}}{(a, x \vee y)_{\rho(\mathrm{WO})}} \qquad \mathrm{OR}_{R_{IOp}} : \frac{(a,x)_{p1}, (b, x)_{p2}, R_{IOp}}{(a \vee b, x)_{\rho(\mathrm{OR})}}$$

We say $(a_1, x_1), \ldots, (a_n, x_n) \vdash_{\mathrm{LDSP}} (a, x)$ if there is a $(a, x)_L$ that can be derived from $\{(a_1, x_1)_0, \ldots, (a_n, x_n)_0\}$.

[3]Phasing has been introduced in a two-phased preference-based deontic logic [6], an extension of Prohairetic Deontic Logic [13] (see the discussion on the logic of imperatives in the previous section) with Hansson's logic DSDL3 [3]. Again this logic differs in several aspects from the logic presented here, for example it contains the axiom (a, a) to facilitate the semantic preference-based presentation.

Before we show the equivalence of the two labelled systems, we again prove lemmas for proof rewriting and consistency checks.

Lemma 12 *For each derivation of $(a, x)_L$ of G in the labelled deductive system with the fulfilment constraint there is an alternative derivation of $(a, x)_L$ of G that respects the order of the labelled deductive system with phasing.*

Proof All possible sequences of two steps of the labelled system that do not respect the order of the labelled deductive system with phasing are given below (left side derivations), together with the alternative derivations that do respect it (right side derivations). Moreover, we show that the label of the root of the left derivation is equivalent to the label of the root of the right derivation, and thus that if the fulfilment constraint hold for the left derivation, then it also holds for the right derivation. This follows from the fact that each element of the label implies the fulfilment of the conditional (Lemma 7) and some elementary operations on $\cup$ and $\times$. For example, consider the two proofs of e.2. concerning a sequence of an OR and a CTA step. The label of the first derived conditional is

$$F_1 \times (F_2 \cup F_3) = (F_1 \times F_2) \cup (F_1 \times F_3)$$

and the label of the second derived conditional is

$$(F_1 \times \{b\} \times F_2) \cup (F_1 \times \{\neg b \vee c\} \times F_3)$$

The two sets contain equivalent formulas, because each element of F_2 classically implies b and each element of F_3 classically implies $\neg b \vee c$ (Lemma 7). The other proofs are analogous.

a. Reversing the order of CTA$_2$ and SI$_1$

$$\dfrac{\dfrac{(a,x)_{F_1} \quad (a \wedge x, y)_{F_2}}{(a, x \wedge y)_{F_1 \times F_2}}\,\text{CTA}}{(a \wedge b, x \wedge y)_{F_1 \times F_2 \times \{b\}}}\,\text{SI} \qquad \dfrac{\dfrac{(a,x)_{F_1}}{(a \wedge b, x)_{F_1 \times \{b\}}}\,\text{SI} \quad \dfrac{(a \wedge x, y)_{F_2}}{(a \wedge b \wedge x, y)_{F_2 \times \{b\}}}\,\text{SI}}{(a \wedge b, x \wedge y)_{F_1 \times F_2 \times \{b\}}}\,\text{CTA}$$

b. Reversing the order of WO$_3$ and SI$_1$

$$\dfrac{\dfrac{(a,x)_F}{(a, x \vee y)_F}\,\text{WO}}{(a \wedge b, x \vee y)_{F \times \{b\}}}\,\text{SI} \qquad \dfrac{\dfrac{(a,x)_F}{(a \wedge b, x)_{F \times \{b\}}}\,\text{SI}}{(a \wedge b, x \vee y)_{F \times \{b\}}}\,\text{WO}$$

c.1. Reversing the order of WO$_3$ and CTA$_2$, case 1

$$\dfrac{\dfrac{(a,x)_{F_1}}{(a, x \vee y)_{F_1}}\,\text{WO} \quad (a \wedge (x \vee y), z)_{F_2}}{(a, (x \vee y) \wedge z)_{F_1 \times F_2}}\,\text{CTA} \qquad \dfrac{\dfrac{(a,x)_{F_1} \quad \dfrac{(a \wedge (x \vee y), z)_{F_2}}{(a \wedge x, z)_{F_2 \times \{x\}}}\,\text{SI}}{(a, x \wedge z)_{F_1 \times F_2 \times \{x\}}}\,\text{CTA}}{(a, (x \vee y) \wedge z)_{F_1 \times F_2 \times \{x\}}}\,\text{WO}$$

c.2. Reversing the order of WO_3 and CTA_2, case 2

$$\frac{(a,x)_{F_1} \quad \dfrac{(a\wedge x,y)_{F_2}}{(a\wedge x,y\vee z)_{F_2}}\,\text{WO}}{(a,x\wedge(y\vee z))_{F_1\times F_2}}\,\text{CTA} \qquad \frac{\dfrac{(a,x)_{F_1} \quad (a\wedge x,y)_{F_2}}{(a,x\wedge y)_{F_1\times F_2}}\,\text{CTA}}{(a,x\wedge(y\vee z))_{F_1\times F_2}}\,\text{WO}$$

d. Reversing the order of OR_4 and SI_1

$$\frac{\dfrac{(a,x)_{F_1} \quad (b,x)_{F_2}}{(a\vee b,x)_{F_1\cup F_2}}\,\text{OR}}{((a\vee b)\wedge c,x)_{(F_1\cup F_2)\times\{c\}}}\,\text{SI} \qquad \frac{\dfrac{(a,x)_{F_1}}{(a\wedge c,x)_{F_1\times\{c\}}}\,\text{SI} \quad \dfrac{(b,x)_{F_2}}{(b\wedge c,x)_{F_2\times\{c\}}}\,\text{SI}}{((a\wedge c)\vee(b\wedge c),x)_{(F_1\times\{c\})\cup(F_2\times\{c\})}}\,\text{OR}$$

e.1. Reversing the order of OR_4 and CTA_2, case 1

$$\frac{\dfrac{(a,x)_{F_1} \quad (b,x)_{F_2}}{(a\vee b,x)_{F_1\cup F_2}}\,\text{OR} \quad ((a\vee b)\wedge x,y)_{F_3}}{(a\vee b,x\wedge y)_{(F_1\cup F_2)\times F_3}}\,\text{CTA}$$

$$\frac{\dfrac{(a,x)_{F_1} \quad \dfrac{((a\vee b)\wedge x,y)_{F_3}}{(a\wedge x,y)_{F_3\times\{a\}}}\,\text{SI}}{(a,x\wedge y)_{F_1\times F_3\times\{a\}}}\,\text{CTA} \quad \dfrac{(b,x)_{F_2} \quad \dfrac{((a\vee b)\wedge x,y)_{F_3}}{(b\wedge x,y)_{F_3\times\{b\}}}\,\text{SI}}{(b,x\wedge y)_{F_2\times F_3\times\{b\}}}\,\text{CTA}}{(a\vee b,x\wedge y)_{(F_1\times F_3\times\{a\})\cup(F_2\times F_3\times\{b\}))}}\,\text{OR}$$

e.2. Reversing the order of OR_4 and CTA_2, case 2

$$\frac{(a,x)_{F_1} \quad \dfrac{(a\wedge x\wedge b,y)_{F_2} \quad (a\wedge x\wedge(\neg b\vee c),y)_{F_3}}{(a\wedge x,y)_{F_2\cup F_3}}\,\text{OR}}{(a,x\wedge y)_{F_1\times(F_2\cup F_3)}}\,\text{CTA}$$

$$\frac{\dfrac{\dfrac{(a,x)_{F_1}}{(a\wedge b,x)_{F_1\times\{b\}}}\,\text{SI} \quad (a\wedge x\wedge b,y)_{F_2}}{(a\wedge b,x\wedge y)_{F_1\times\{b\}\times F_2}}\,\text{CTA} \quad \dfrac{\dfrac{(a,x)_{F_1}}{(a\wedge(\neg b\vee c),x)_{F_1\times\{\neg b\vee c\}}}\,\text{SI} \quad (a\wedge x\wedge(\neg b\vee c),y)_{F_3}}{(a\wedge(\neg b\vee c),x\wedge y)_{F_1\times\{\neg b\vee c\}\times F_3}}}{(a,x\wedge y)_{(F_1\times\{b\}\times F_2)\cup(F_1\times\{\neg b\vee c\}\times F_3)}}\,\text{OR}$$

f. Reversing the order of OR_4 and WO_3

$$\frac{\dfrac{(a,x)_{F_1} \quad (b,x)_{F_2}}{(a\vee b,x)_{F_1\cup F_2}}\,\text{OR}}{(a\vee b,x\vee y)_{F_1\cup F_2}}\,\text{WO} \qquad \frac{\dfrac{(a,x)_{F_1}}{(a,x\vee y)_{F_1}}\,\text{WO} \quad \dfrac{(b,x)_{F_2}}{(b,x\vee y)_{F_2}}\,\text{WO}}{(a\vee b,x\vee y)_{F_1\cup F_2}}\,\text{OR}$$

■

The second lemma shows that for phased proofs a consistency check on the fulfilments is equivalent with a consistency check on the conjunction of the input and output of the phase 1 conditionals.

Lemma 13 *Consider any potential derivation of the labelled system with phasing, satisfying R_p but not necessarily R_{IO}, as well as its counterpart in the labelled system with the fulfilment constraint, not necessarily satisfying R_F. Then the following four conditions are equivalent:*

1. *The first derivation satisfies condition R_{IO} throughout phase 1,*
2. *The first derivation satisfies condition R_{IO} everywhere.*
3. *The second derivation satisfies condition R_F throughout phase 1,*
4. *The second derivation satisfies R_F everywhere,*

Proof *Clearly $(2) \Rightarrow (1)$ and $(4) \Rightarrow (3)$. Through phase 1, for each formula in the derivation with fulfilments the conjunction of input and output is equivalent to the unique element of its label. Hence $(1) \Leftrightarrow (3)$. In phase 2 the conjunction of input and output is also equivalent to the unique element of its label, which is equivalent to the label of the second premise of each derivation step. In phase 3 and 4 the rules preserve the consistency of the conjunction of input and output, and they also preserve the property that each element of the label is consistent. From this we have $(1) \Rightarrow (2)$ and $(3) \Rightarrow (4)$. Putting this together gives us $(1) \Leftrightarrow (2) \Leftrightarrow (3) \Leftrightarrow (4)$ and we are done.*

In Theorem 14 below we show that for each derivation in the labelled deductive system with the fulfilment constraint there is an equivalent derivation in the labelled deductive system with phasing.

Theorem 14 (Equivalence LDSF **and** LDSP**)** *Let G be a consistent conditional base. We have $G \vdash_{\text{LDSF}} (a, x)$ if and only if $G \vdash_{\text{LDSP}} (a, x)$.*

Proof For each proof of (a, x) from G in one system, we can give a proof in the other system. First we show how a proof tree of the second system can be constructed from the proof tree of the first system, and then we show that if the fulfilment constraint holds for the first proof, then it must also hold for the second proof.

$\Rightarrow$ Given a proof of (a, x) in the labelled deductive system with the fulfilment constraint. To construct a proof in the labelled deductive system with the phasing constraint, first iteratively replace two subsequent steps in the wrong order by several steps in the right order, see Lemma 12. For a finite tree, after a finite number of steps all derivation steps are ordered, because no set of replacements cycles (and can be used to construct infinite proof trees). Then label the premises with 0, and label the other nodes as indicated by the rules of the labelled system. It follows from Lemma 13 that if the consistency constraint holds for the first proof, then it also holds for the second proof.

$\Leftarrow$ Given a proof of (a, x) in the labelled deductive system with the phasing constraint. To construct a proof in the labelled deductive system with the fulfilment constraint, label the premises with the conjunction of input and output, and label the other nodes as indicated by the rules of the labelled system. It follows from Lemma 13 that if the consistency constraint holds for the first proof, then it also holds for the second proof. ■

Corollary 15 *Let G be a consistent conditional base. We have $G \vdash_{\mathrm{ROF}} (a, x)$ if and only if $G \vdash_{\mathrm{LDSP}} (a, x)$.*

Proof Follows directly from Theorem 10 and 14. ■

6. PHASING: SOME EXAMPLES

In this final section we illustrate phasing by several examples. To emphasize the intuitions behind phasing, we introduce some new terminology. We call phase-1, phase-2 and phase-4 conditionals respectively atomic, compound and combined conditionals. Note that the logic of imperatives covers compounding but not combining; the other logics discussed in this paper cover both.

Definition 16 *Let G be a consistent conditional base.*

- *The conditional (a, x) is an* atomic conditional *of G if $a \wedge x$ is consistent and there is a conditional $(b, x) \in G$ such that $a \vdash b$.*
- *The conditional $(a, x_1 \wedge \ldots \wedge x_n)$ is a* compound conditional *of G if there is a set $\{(a, x_1), (a \wedge x_1, x_2), \ldots, (a \wedge x_1 \wedge \ldots \wedge x_{n-1}, x_n)\}$ of atomic conditionals of G.*
- *The conditional $(a_1 \vee \ldots \vee a_n, x)$ is a* combined conditional *of G if there is a set $\{(a_1, x_1), \ldots, (a_n, x_n)\}$ of compound conditionals of G such that $x_i \vdash x$ for $i = 1 \ldots n$.*

Finally, all sets of conditionals are closed under logical equivalence for input and output.

The new terminology is illustrated by the following example, inspired from preference-based reasoning.[4] Consider the set of atomic propositions a_i for

[4]In preference-based reasoning compounding and combining correspond to the construction of a preference ordering and minimization in this constructed ordering [6, 7].

$i = 1, \ldots, n$ and $x_{i,j}$ for $i = 1, \ldots, n$ and $j = 1, \ldots, m_n$. We write b_i when a_i is true and the other a_j are false, and we write $y_{i,j}$ when b_i is true as well as the $x_{i,1}, \ldots, x_{i,j}$.

$$\begin{array}{lll} b_i & =_{\mathit{def}} & a_i \wedge \bigwedge_{j \neq i} \neg a_j \\ y_{i,j} & =_{\mathit{def}} & b_i \wedge x_{i,1} \wedge \ldots \wedge x_{i,j}. \end{array}$$

Now consider the following set of atomic conditionals.

$$G = \left\{ \begin{array}{cccc} (b_1, y_{1,1}), & (b_2, y_{2,1}), & \ldots, & (b_n, y_{n,1}) \\ (y_{1,1}, y_{1,2}), & (y_{2,1}, y_{2,2}), & \ldots, & (y_{n,1}, y_{n,2}) \\ \ldots, & \ldots, & & \ldots \\ (y_{1,m_1-1}, y_{1,m_1}), & (y_{2,m_2-1}, y_{2,m_2}), & \ldots, & (y_{n,m_n-1}, y_{n,m_n}) \end{array} \right\}$$

Each column of atomic conditionals can be used to derive the compound conditional (b_i, y_{i,m_i}), and together they can be used to derive the combined conditional $(b_1 \vee \ldots \vee b_n, y_{1,m_1} \vee \ldots \vee y_{1,m_n})$. This derivation is illustrated in Figure 1.1. Each column is a case: compounding is reasoning within a case and combining is reasoning by cases.

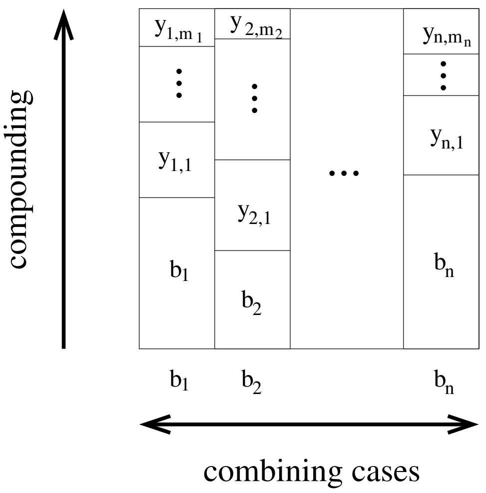

Figure 1.1 Compounding and combining

Before we discuss more examples, we show the straightforward equivalence of the new phasing definition with the labelled deductive system with phasing. The following lemma shows that in the labelled system with phasing the successive applications of the same proof rule can be replaced by a single application of a generalized rule.

Lemma 17 *A set of successive applications of* SI *can be replaced by single* SI *step, and a set of successive applications of* WO *can be replaced by single* WO *step. A set of successive applications of* CTA *can be replaced by the following generalized cut rule, and a set of successive applications of* OR *can be replaced by the following generalized disjunction rule.*

$$\mathrm{GCTA}_{R_{IOp}} : \frac{(a, x_1)_{p_1}, \ldots, (a \wedge x_1 \wedge \ldots \wedge x_{n-1}, x_n)_{p_n}, R_{IOp}}{(a, x_1 \wedge \ldots \wedge x_n)_{\rho(\mathrm{CTA})}}$$

$$\mathrm{GOR}_{R_{IOp}} : \frac{(a_1, x)_{p_1}, \ldots, (a_n, x)_{p_n}, R_{IOp}}{(a_1 \vee \ldots \vee a_n, x)_{\rho(\mathrm{OR})}}$$

Proof For SI and WO the proofs are trivial. For CTA, it follows from the fact that the order of CTA steps is irrelevant, as an inductive generalization of the case for three premises below.

$$\dfrac{(a, x)_{p_1} \qquad \dfrac{(a \wedge x, y)_{p_2} \quad (a \wedge x \wedge y, z)_{p_3}}{(a \wedge x, y \wedge z)_{\rho(\mathrm{CTA})}}\ \mathrm{CTA}}{(a, x \wedge y \wedge z)_{\rho(\mathrm{CTA})}}\ \mathrm{CTA}$$

$$\dfrac{\dfrac{(a, x)_{p_1} \quad (a \wedge x, y)_{p_2}}{(a, x \wedge y)_{\rho(\mathrm{CTA})}}\ \mathrm{CTA} \qquad (a \wedge x \wedge y, z)_{p_3}}{(a, x \wedge y \wedge z)_{\rho(\mathrm{CTA})}}\ \mathrm{CTA}$$

Similarly, for OR the proof follows from the fact that the order of OR steps is irrelevant.

$$\dfrac{(a, x)_{p_1} \qquad \dfrac{(b, x)_{p_2} \quad (c, x)_{p_3}}{(b \vee c, x)_{\rho(\mathrm{OR})}}\ \mathrm{OR}}{(a \vee b \vee c, x)_{\rho(\mathrm{OR})}}\ \mathrm{OR}$$

$$\dfrac{\dfrac{(a, x)_{p_1} \quad (b, x)_{p_2}}{(a \vee b, x)_{\rho(\mathrm{OR})}}\ \mathrm{R} \qquad (c, x)_{p_3}}{(a \vee b \vee c, x)_{\rho(\mathrm{OR})}}\ \mathrm{R}$$

■

This gives us the final theorem of this paper.

Theorem 18 (Equivalence LDSP **and phasing)** *Let G be a consistent conditional base. We have* $G \vdash_{\mathrm{LDSP}} (a, x)$ *if and only if* (a, x) *is a combined conditional of G.*

Proof Lemma 17 shows how a proof in the deductive system with phasing can be rewritten into a phased proof tree of depth of at most four. From Lemma 13 follows that a consistency check on all steps is equivalent to a consistency check on atomic conditionals only, and we are done. ■

As a further illustration of phasing we consider the conditional bases discussed in Section 2. and 3. We assume – without mentioning it explicitly – that the sets also contain conditionals where inputs and outputs are replaced by logical equivalents, and we write $cons(a)$ for the consistency check in $\mathcal{L}$ on a. The first base $G = \{(a, b), (b, c), (c, d)\}$ illustrates that derivations go as far as possible by compounding atomic conditionals.

- atomic conditionals are (a, b), $(a \wedge b, c)$ and $(a \wedge b \wedge c, d)$;
- a compound conditional is $(a, b \wedge c \wedge d)$;
- a combined conditional is (a, d).

The second conditional base $G = \{(a, x \wedge y), (\neg a, x \wedge \neg y)\}$ illustrates that reasoning by cases is done by combining compound conditionals.

- the atomic and compound conditionals are

$$\{(a \wedge b, x \wedge y) \mid cons(a \wedge b \wedge x \wedge y)\}$$

$$\{(\neg a \wedge b, x \wedge \neg y) \mid cons(\neg a \wedge b \wedge x \wedge \neg y)\}$$

- the combined conditionals are

$$\{(a \wedge b, (x \wedge y) \vee z) \mid cons(a \wedge b \wedge x \wedge y)\}$$

$$\{(\neg a \wedge b, (x \wedge \neg y) \vee z) \mid cons(\neg a \wedge b \wedge x \wedge \neg y)\}$$

$$\{(\top, x \vee z) \mid cons(a \wedge b \wedge x \wedge y) \wedge cons(\neg a \wedge b \wedge x \wedge \neg y)\}$$

The third base $G = \{(\top, c), (c, i), (\neg c, \neg i)\}$ illustrates context-sensitivity, because $(\neg c, i)$ is not a conditional of G. The conditional $(\top, i)$ is a compound conditional but not an atomic conditional; therefore it cannot be used to derive this counterintuitive conditional. Note that the combined conditionals $\{(a \wedge c, i \vee x) \mid cons(a \wedge c \wedge i)\})$ are implied by the combined conditionals $\{(a, (c \wedge i) \vee x) \mid cons(a \wedge c \wedge i)\}$.

- the atomic conditionals are

$$\{(a, c) \mid cons(a \wedge c)\}$$

$$\{(a \wedge c, i) \mid cons(a \wedge c \wedge i)\}$$

$$\{(a \wedge \neg c, \neg i) \mid cons(a \wedge \neg c \wedge \neg i)\}$$

- the compound conditionals are

$$\{(a, c) \mid cons(a \wedge c)\}$$

$$\{(a \wedge c, i) \mid cons(a \wedge c \wedge i)\}$$

$$\{(a, c \wedge i) \mid cons(a \wedge c \wedge i)\}$$

$$\{(a \wedge \neg c, \neg i) \mid cons(a \wedge \neg c \wedge \neg i)\}$$

- the combined conditionals are

$$\{(a, c \vee x) \mid cons(a \wedge c)\}$$

$$\{(a, (c \wedge i) \vee x) \mid cons(a \wedge c \wedge i)\}$$

$$\{(a \wedge \neg c, \neg i \vee x) \mid cons(a \wedge \neg c \wedge \neg i)\}$$

The fourth base $G = \{(\top, x), (\top, \neg x)\}$ illustrates conflict tolerance, because $(\top, y)$ is not a conditional of G if y is not classically implied by x or $\neg x$. The conditional $(\top, x \vee y)$ is a combined conditional but not an atomic conditional; therefore it cannot be used to derive this counterintuitive conditional.

- the atomic and compound conditionals are

$$\{(a, x) \mid cons(a \wedge x)\}$$

$$\{(a, \neg x) \mid cons(a \wedge \neg x)\}$$

- the combined conditionals are

$$\{(a, x \vee y) \mid cons(a \wedge x)\}$$

$$\{(a, \neg x \vee y) \mid cons(a \wedge \neg x)\}$$

The examples illustrate that for small bases (such as the lists of benchmark examples in [4] and [13]) it is much easier to write the set of all combined conditionals than the set of all implied conditionals of the logic of reusable propositional output with the fulfilment constraint. Surprisingly, we have shown in this paper that (a, x) can be derived from the base G if and only if there is a combined conditional for (a, x) of G.

Acknowledgment

Thanks to Patrick van der Laag, David Makinson and Emil Weydert for discussions on the issues raised in this paper, and comments on earlier versions of it. Many of the issues and examples discussed in this paper were raised during discussions with David Makinson.

References

[1] R.M. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24:33–36, 1963.

[2] D. Gabbay. *Labelled Deductive Systems*, volume 1. Oxford University Press, 1996.

[3] B. Hansson. An analysis of some deontic logics. In R. Hilpinen, editor, *Deontic Logic: Introductory and Systematic Readings*, pages 121–147. D. Reidel Publishing Company, Dordrecht, Holland, 1971. A reprint from *Noûs*, 1969.

[4] D. Makinson. On a fundamental problem of deontic reasoning. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems. New Studies on Deontic Logic and Computer Science*, pages 29–53. IOS Press, 1999.

[5] D. Makinson and L. van der Torre. The logic of reusable propositional output. 1999. Submitted.

[6] Y. Tan and L. van der Torre. How to combine ordering and minimizing in a deontic logic based on preferences. In *Deontic Logic, Agency and Normative Systems. Proceedings of the ΔEON'96*, Workshops in Computing, pages 216–232. Springer, 1996.

[7] L. van der Torre. *Reasoning about Obligations: Defeasibility in Preference-based Deontic Logic*. PhD thesis, Erasmus University Rotterdam, 1997.

[8] L. van der Torre. Labeled logics of conditional goals. In *Proceedings of the Thirteenth European Conference on Artificial Intelligence (ECAI'98)*, pages 368–369. John Wiley & Sons, 1998.

[9] L. van der Torre. Phased labeled logics of conditional goals. In *Logics in Artificial Intelligence*, LNAI 1489, pages 92–106. Springer, 1998.

[10] L. van der Torre. Defeasible goals. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, LNAI 1638. Springer, 1999.

[11] L. van der Torre and Y. Tan. The many faces of defeasibility in defeasible deontic logic. In D. Nute, editor, *Defeasible Deontic Logic*, volume 263 of *Synthese Library*, pages 79–121. Kluwer, 1997. Extended and revised version of L. van der Torre and Y. Tan, Cancelling and overshadowing: two types of defeasibility in defeasible deontic logic. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI'95)*, pages 1525–1532. Morgan Kaufman, 1995.

[12] L. van der Torre and Y. Tan. Diagnosis and decision making in normative reasoning. *Artificial Intelligence and Law*, 7:51–67, 1999. Revised and extended version of L. van der Torre and Y. Tan. Distinguishing different roles in normative reasoning. In *Proceedings of the Sixth International Conference on AI and Law (ICAIL'97)*, pages 225–232. ACM Press, 1997.

[13] L. van der Torre and Y. Tan. Contrary-to-duty reasoning with preference-based dyadic obligations. *Annals of Mathematics and Artificial Intelligence*, 2000. To appear. Revised and extended version of L. van der Torre and Y. Tan. Prohairetic Deontic Logic (PDL). In *Logics in Artificial Intelligence*, LNAI 1489, pages 77-91. Springer, 1998.

[14] L. van der Torre and E. Weydert. Goals, desires, utilities and preferences. In *Proceedings of the ECAI'98 Workshop on Decision Theory meets Artificial Intelligence*, 1998.

[15] L. van der Torre and E. Weydert. Parameters for utilitarian desires in a qualitative decision theory. 1999. Submitted. Revised and extended version of L. van der Torre and E. Weydert. Parameters for utilitarian desires (extended abstract). In *Proceedings of the IJCAI'99 Workshop on Practical Reasoning and Rationality (PRR'99)*, 1999.

[16] G.H. von Wright. A new system of deontic logic. In R. Hilpinen, editor, *Deontic Logic: Introductory and Systematic Readings*, pages 105–120. D. Reidel Publishing Company, Dordrecht, Holland, 1971. A reprint of 'A New System of Deontic Logic,' *Danish Yearbook of Philosophy* 1:173–182, 1964, and 'A Correction to a New System of Deontic Logic', *Danish Yearbook of Philosophy* 2:103–107, 1965.